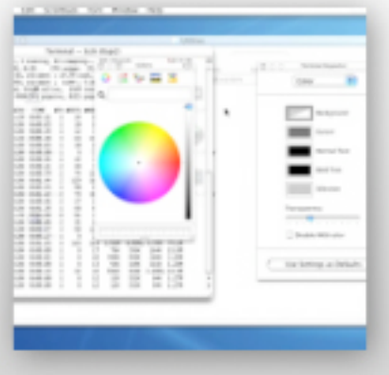
Kerberos in Mac OS X

Session 108



Kerberos in Mac OS X

Craig Keithley
Security Technology Evangelist
keithley@apple.com

Kerberos Overview

- What is Kerberos?
- How Kerberos works
- Kerberos infrastructures
- New features of Kerberos in Panther
- How to add Kerberos to your application
- Summary and wrap up
- Questions and answers

What Is Kerberos?

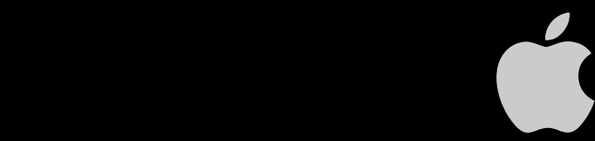
- A secure network authentication system
  - Authentication versus Authorization
- Proves who you are
- Uses a trusted third-party security model

Who Uses Kerberos?

- Government
- Large corporations
- Higher education
- Every Windows 2000 (or later) installation
- Mac OS X 10.1 and later

Why Do They Use It?

- Factors security problem
  (Authentication vs. Authorization)
- Consistent with centralized authentication goals
- Single sign-on
- Ubiquitous
  - Windows, Solaris, Mac OS X, Linux, AIX, Xbox Live…
- Time tested
- Source freely available



Kerberos for Macintosh

Marshall Vale
Massachusetts Institute of Technology
mjv@mit.edu

Brief Kerberos History

- Developed in the 1980s as part of MIT's Project Athena
- IETF proposed standard in 1993
- Selected as the default authentication method in Windows 2000 and later
- Selected by Apple as the default authentication method in Mac OS X Panther and later
- Large increase of use in the enterprise market

Kerberos Protocols

- Kerberos v4
  - Original version, no longer developed
  - Still in use at many long-standing Kerberos sites
- Kerberos v5
  - Current version of the Kerberos protocol
  - Designed to solve many deficiencies and security problems in Kerberos v4

Kerberos Terms

- Realm
  - The administrative unit protected by a Key Distribution Center (KDC)
- Ticket
  - An authentication token
  - Different types of tickets for different purposes
  - Have a finite lifetime; they expire
  - Also known as “credentials”

Kerberos Terms

- Ticket-granting ticket (TGT)
  - An initial ticket that grants permission to get service tickets
- Service ticket
  - A ticket that authenticates the user to a particular service (for instance, e-mail, ftp, login)

Realm

- Looks similar to an Internet domain
  - ATHENA.MIT.EDU
- Defines your namespace
- Unlike DNS, a Kerberos realm is case sensitive
- Can have trust relationships between realms
- Can support any combination of v4 and v5
- As few realms as possible in an organization is best

Ticket-Granting Ticket (TGT)

- Also called “initial ticket” because it is the first ticket a client gets
- Proves that the client is allowed to get tickets for other services
- Acts as a substitute for password
- Mechanism by which single sign on is achieved
- Only valid for a limited period of time (expires), although can be renewable

Getting a TGT

Client asks KDC for TGT

Getting a TGT

KDC returns a TGT and session key to a client encrypted with client's key

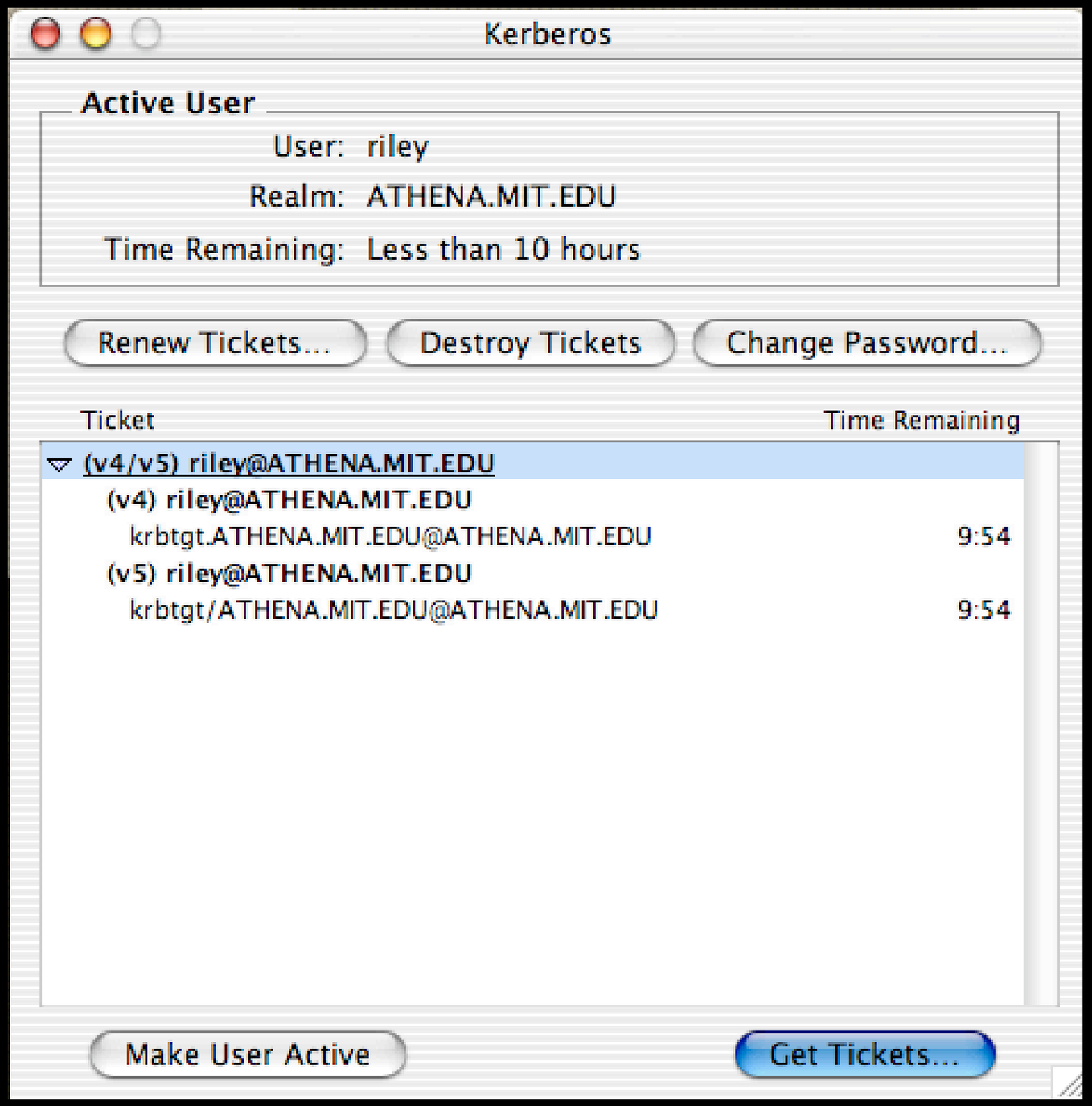
Getting a TGT

Client uses client's key (one-way hash of password) to extract TGT

KDC

Client
TGT

Getting a TGT

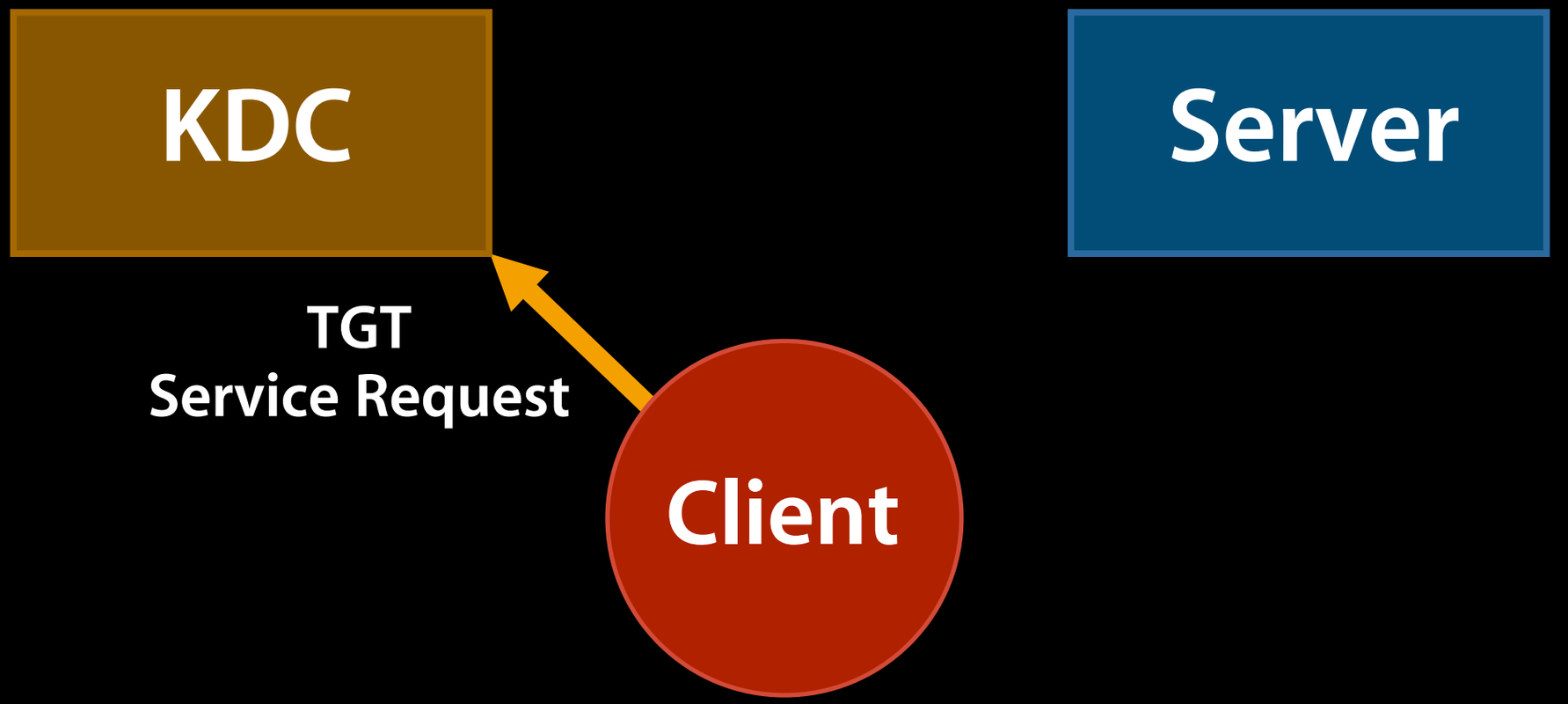


Service Ticket

- Ticket used by client to get access to particular service or server (for instance, ftp, mail, ssh)
- Contains a session key that is shared between the client and server that they can use to encrypt data exchanges
- Also expires

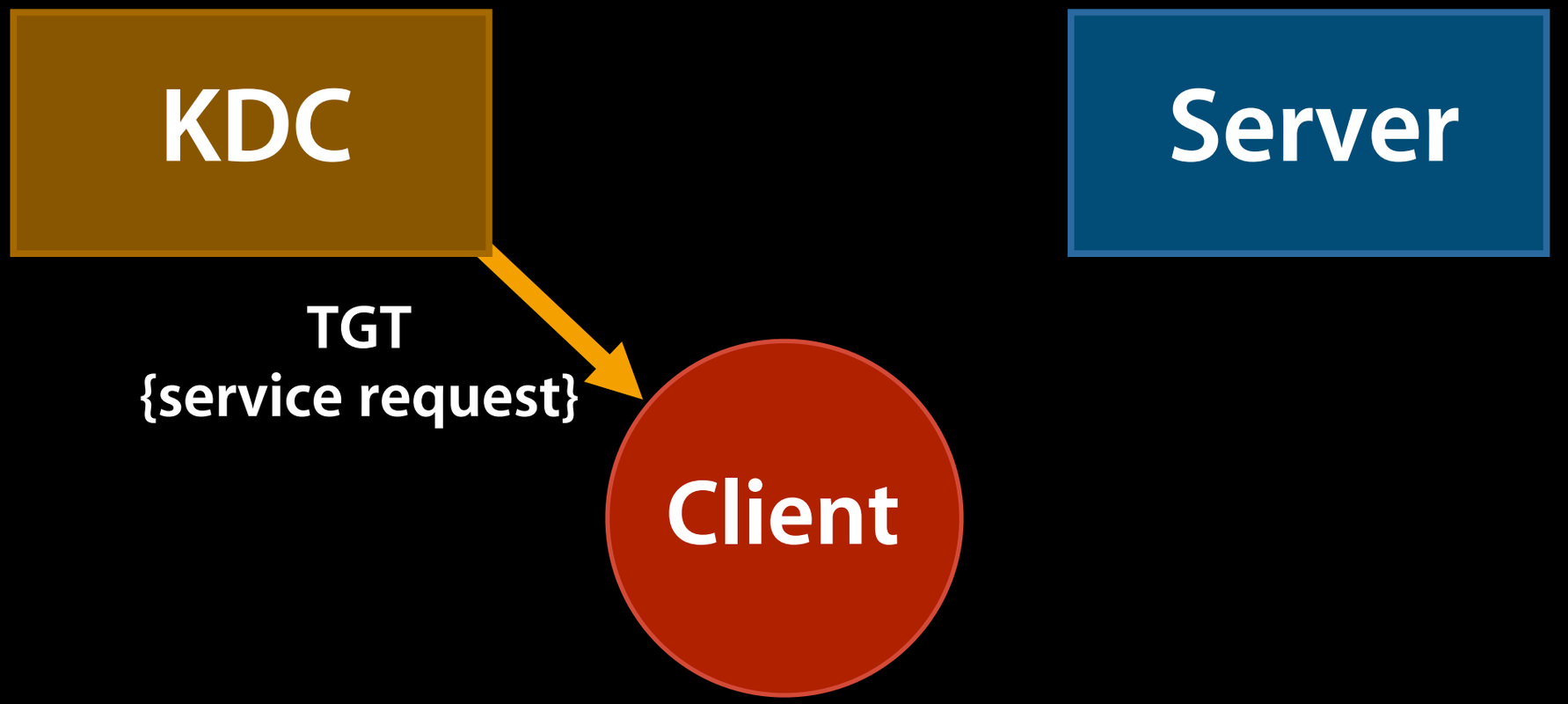
Getting a Service Ticket

- Client presents TGT to KDC with a request for a service ticket



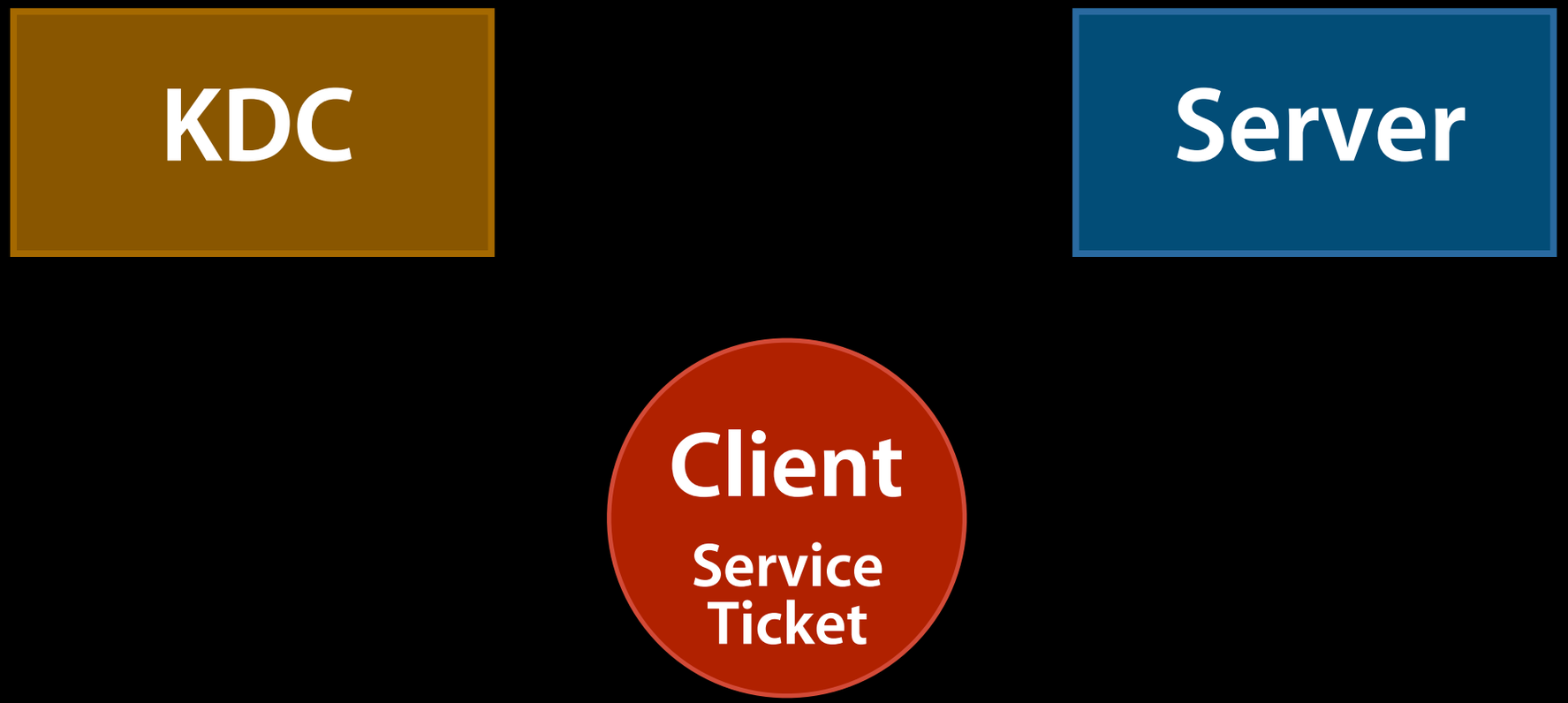
Getting a Service Ticket

- KDC returns encrypted service ticket to client



Getting a Service Ticket

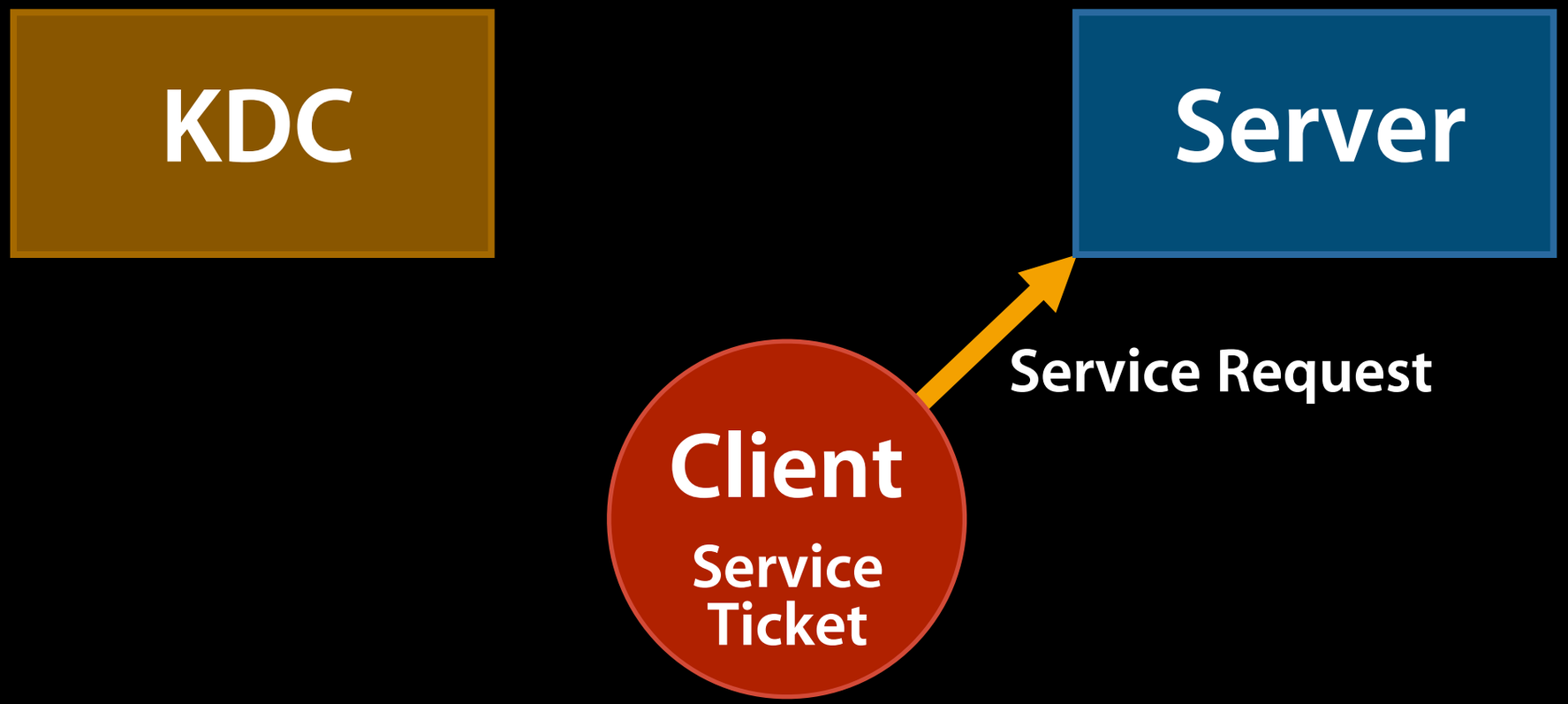
- Client uses TGT to extract service ticket



Connecting to a Service

One-way authentication

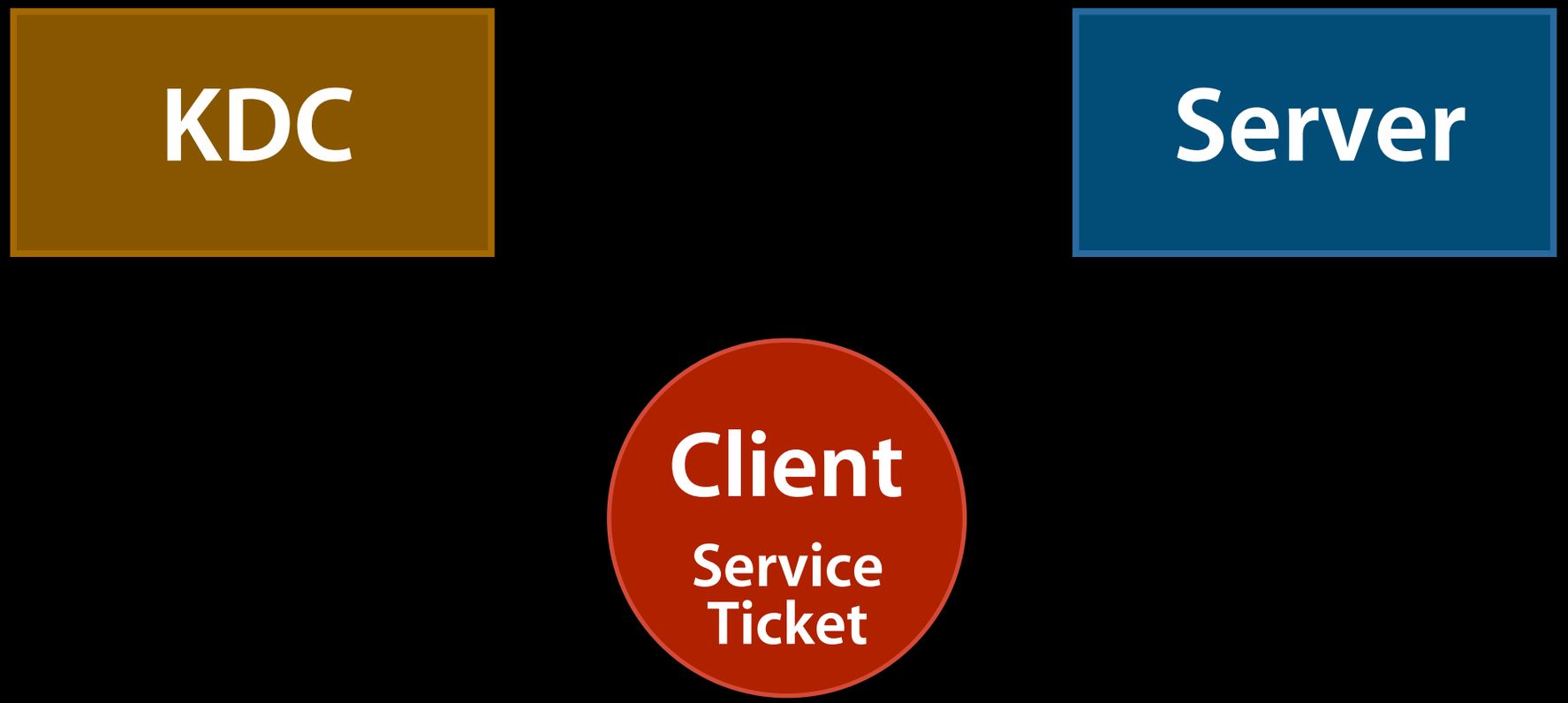
- Client presents service ticket to server



Connecting to a Service

One-way authentication

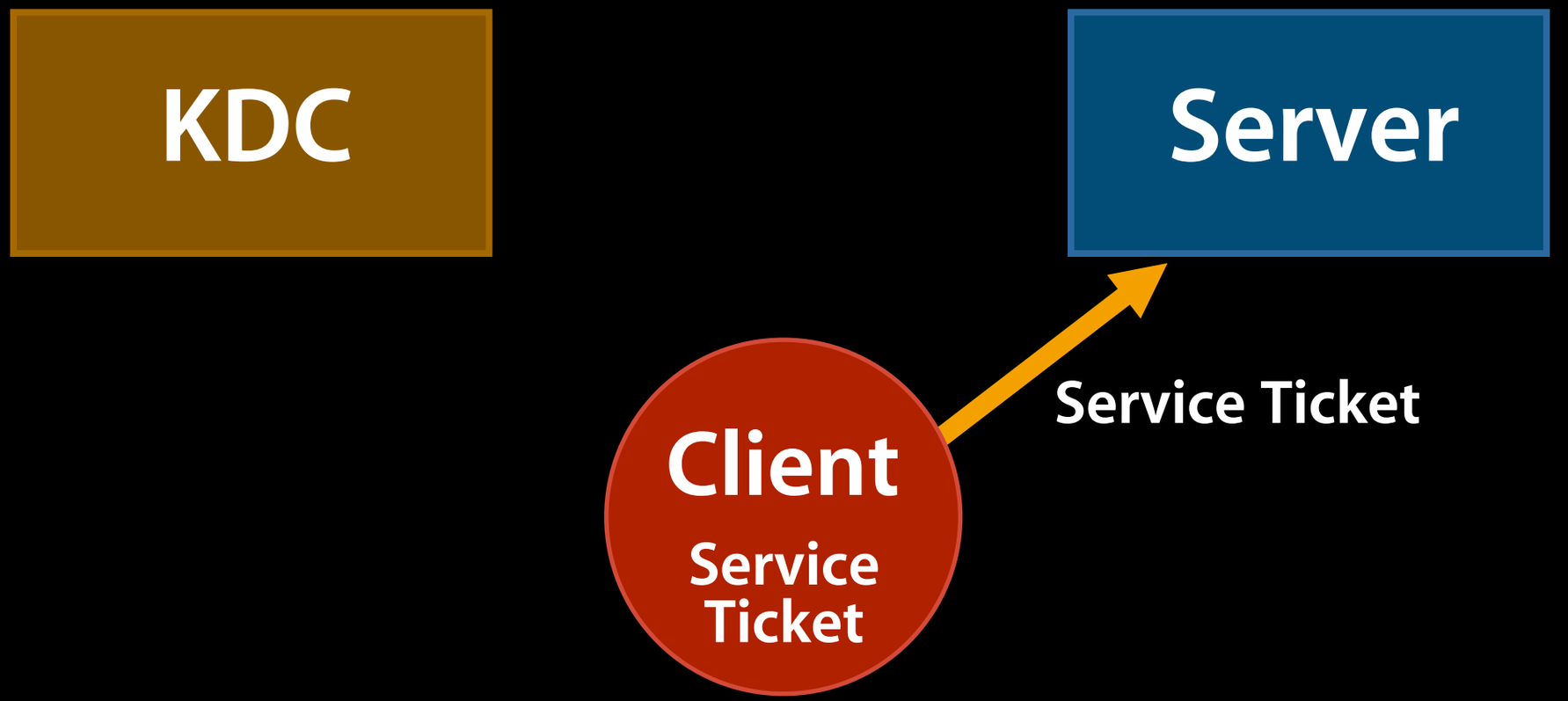
- Server receives service ticket and authenticates the client
- Server should use session key from service ticket to encrypt subsequent traffic between client and server



Connecting to a Service

Mutual authentication

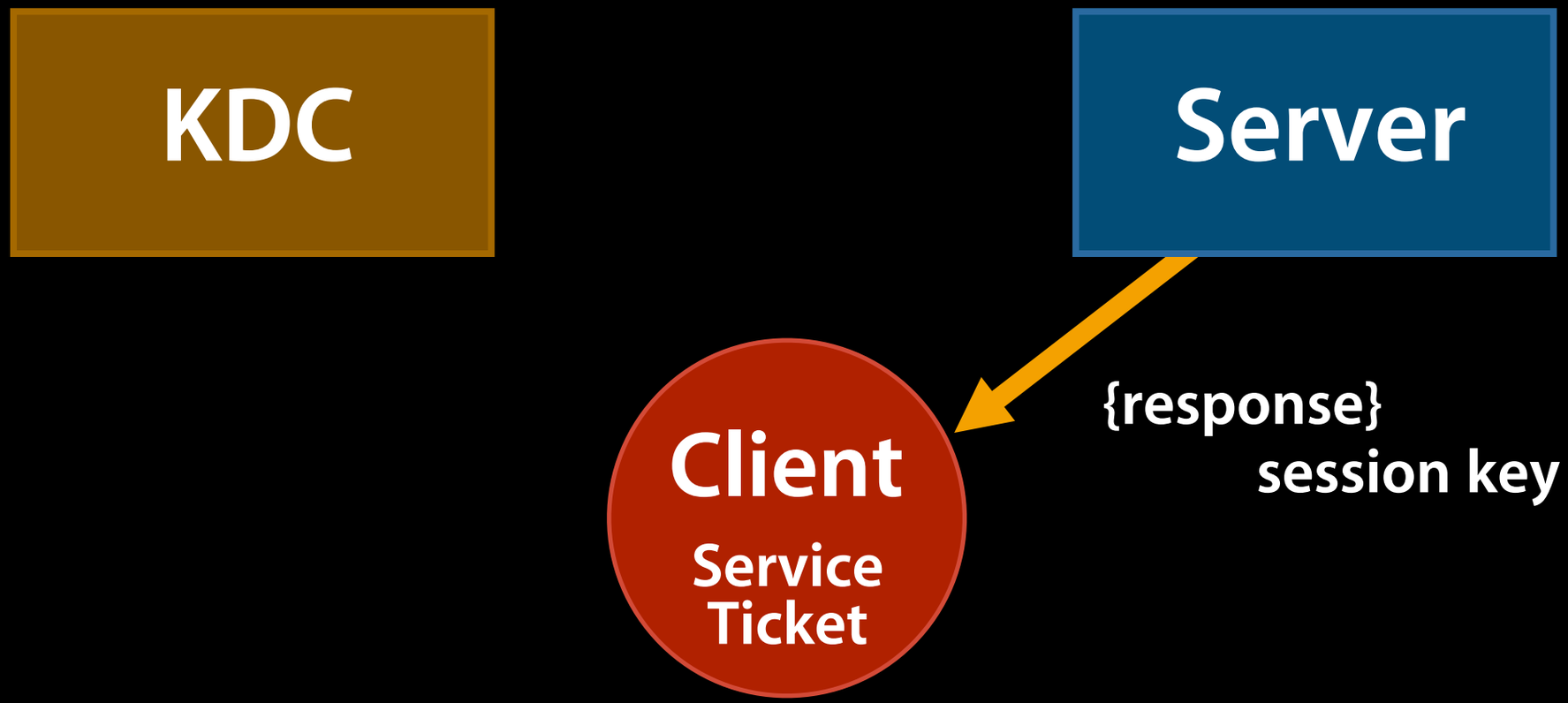
- Client sends service ticket containing session key to server



Connecting to a Service

Mutual authentication

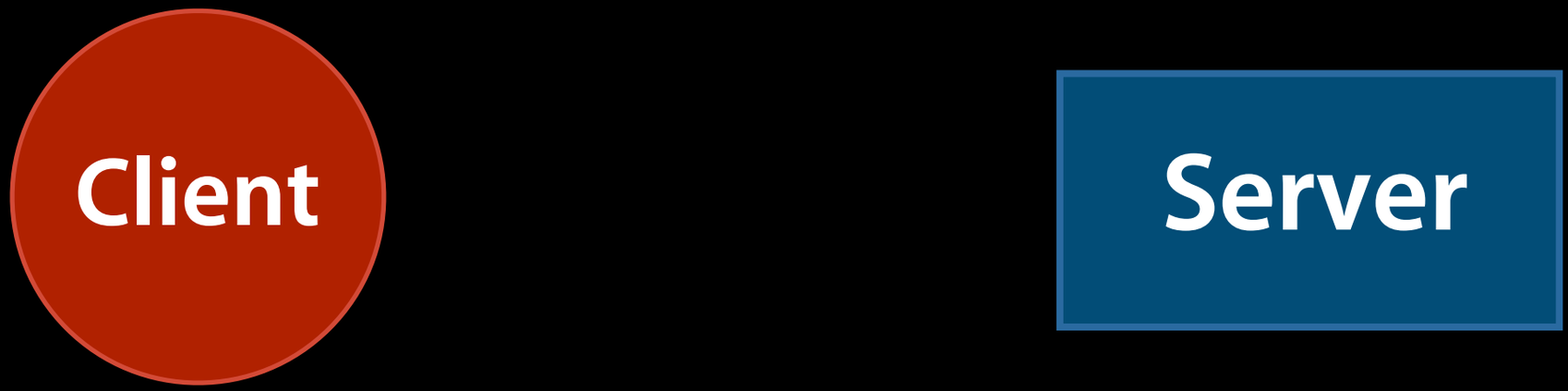
- Server returns a response encrypted with session key from service ticket, authenticating server to client
- Server should also use session key to encrypt subsequent traffic between client and server



Connecting to a Service

IMAP example

- Uses Simple Authentication and Security Layer (SASL) for security
- Use of SASL implies GSSAPI for Kerberos v5
- Base 64 encoded protocol data



Connecting to a Service

IMAP example

- Client requests server capabilities

Connecting to a Service

IMAP example

- Server lists capabilities including GSSAPI

Connecting to a Service

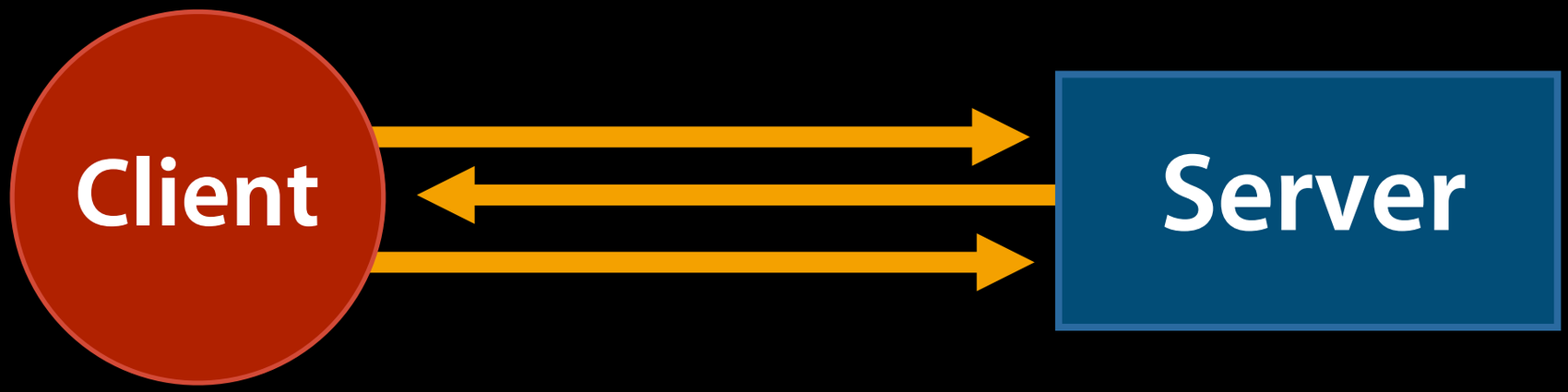
IMAP example

- Client selects GSSAPI

Connecting to a Service

IMAP example

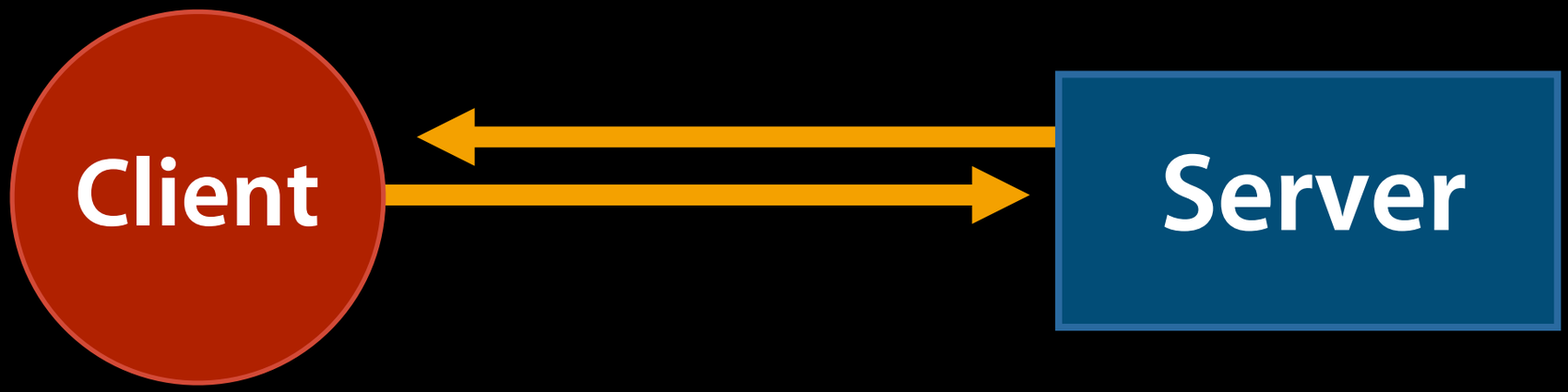
- Client selects GSSAPI
- Kerberos authenticator sent as part of a GSSAPI token



Connecting to a Service

IMAP example

- Server sends mutual authentication response as part of a GSSAPI token
- Client responds with an acknowledgment



Connecting to a Service
IMAP example

- Server sends back security options (buffer size, whether encryption should be used)

Connecting to a Service

IMAP example

- Clients sends who it wants to log in as and selects encryption

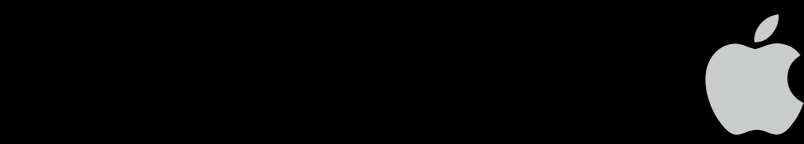
Connecting to a Service

IMAP example

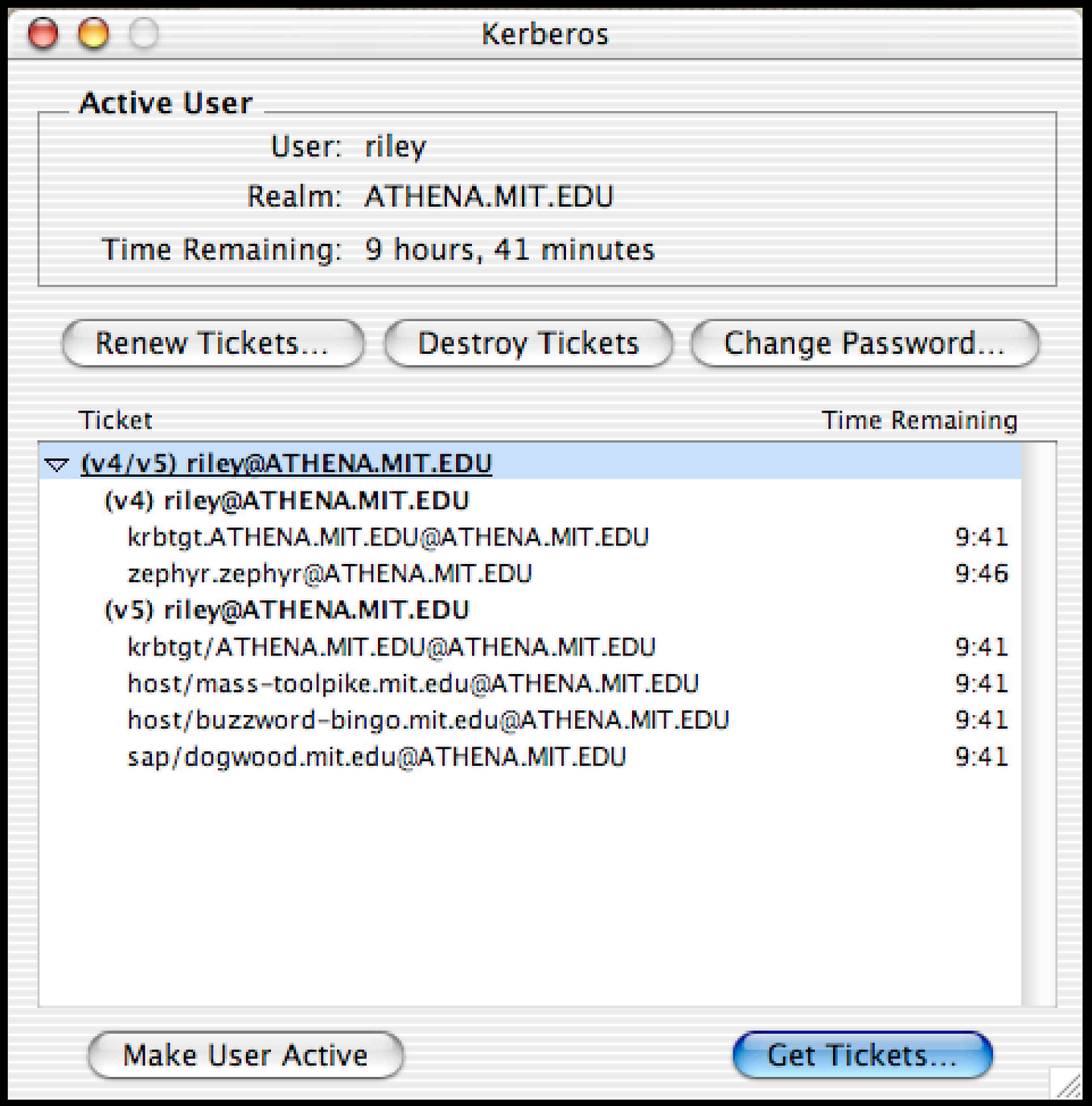
- Server sends back whether the client was authorized and whether encryption was selected

Connecting to a Service

- Base protocol proceeds as normal with encryption
- Can use session key to negotiate encryption keys
- All ticket exchanges are invisible to the user
- Connecting to a service is automatic
  - No need to type password in again
  - Negotiation of Kerberos features automatic
- Allows for single sign-on



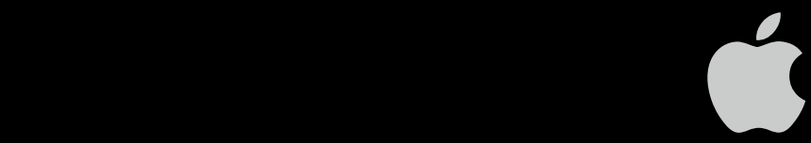
Connecting to a Service



Kerberos Infrastructure

Kerberos Infrastructure

- Three styles of KDC setups
  - MIT KDC
  - MIT KDC with Microsoft Active Directory
  - Apple KDC
- General KDC recommendations



MIT KDC

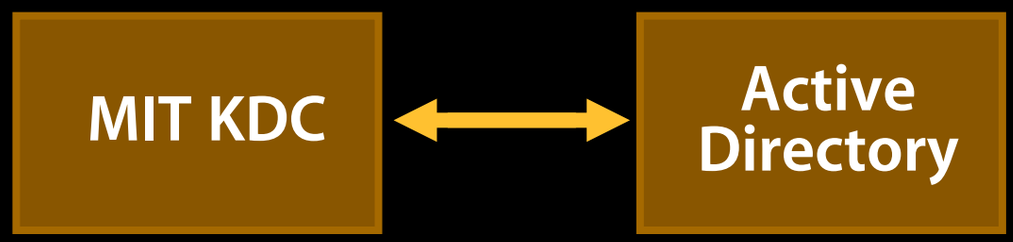
- Provides best support for cross-platform environments
- Provides flexibility for special deployments such as AFS
- Complex to manage and configure
- Source code available

MIT KDC With Active Directory

- Provides good support for Windows integration
- There is a cross domain trust between AD and MIT KDC
- MIT KDC contains
  - Canonical user accounts
  - Host keys for non-Windows machines and services
- There are mappings of user accounts in MIT KDC to accounts in AD Domain

MIT KDC With Active Directory

Account mapping



Non-Windows Application Service

Windows Application Service

User

Apple KDC

- Easy to set up and configure
- Takes advantage of Open Directory
- Password Server contains Kerberos keys and other passwords for legacy client support
- Machines talk to the directory to get configuration and account information
- Machines talk to the Apple KDC for Kerberos authentication

General KDC Recommendations

- Use Tripple DES or RC4 instead of DES for all keys
- Set the preauth_required flag on user principals
- Run as few other services as possible on your KDC
- Avoid Kerberos v4
- Deploy client configuration files with as few settings explicitly set as possible
- Securely backup your KDC

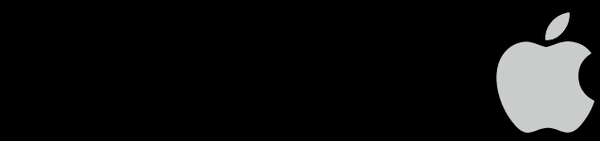
AFS Considerations

- Andrew File System uses Kerberos v4
- Use aklog found at http://mit.edu/openafs to get tokens
- Run a krb524d on your KDC
- Only use Kerberos v4 for the AFS principal; do not globally enable Kerberos v4

Kerberos for Macintosh

Jaguar Kerberos

- Included Kerberos for Macintosh 4.5
  - Kerberos ticket management application bundled
  - Kerberos v5 1.2.6
  - Performance enhancements
  - Improvements to command line utilities
  - 524 support
- Kerberos for Macintosh 4.5.1
  - Released with 10.2.3 software update



Panther Kerberos

Kerberos for Macintosh 5.0

New features include:

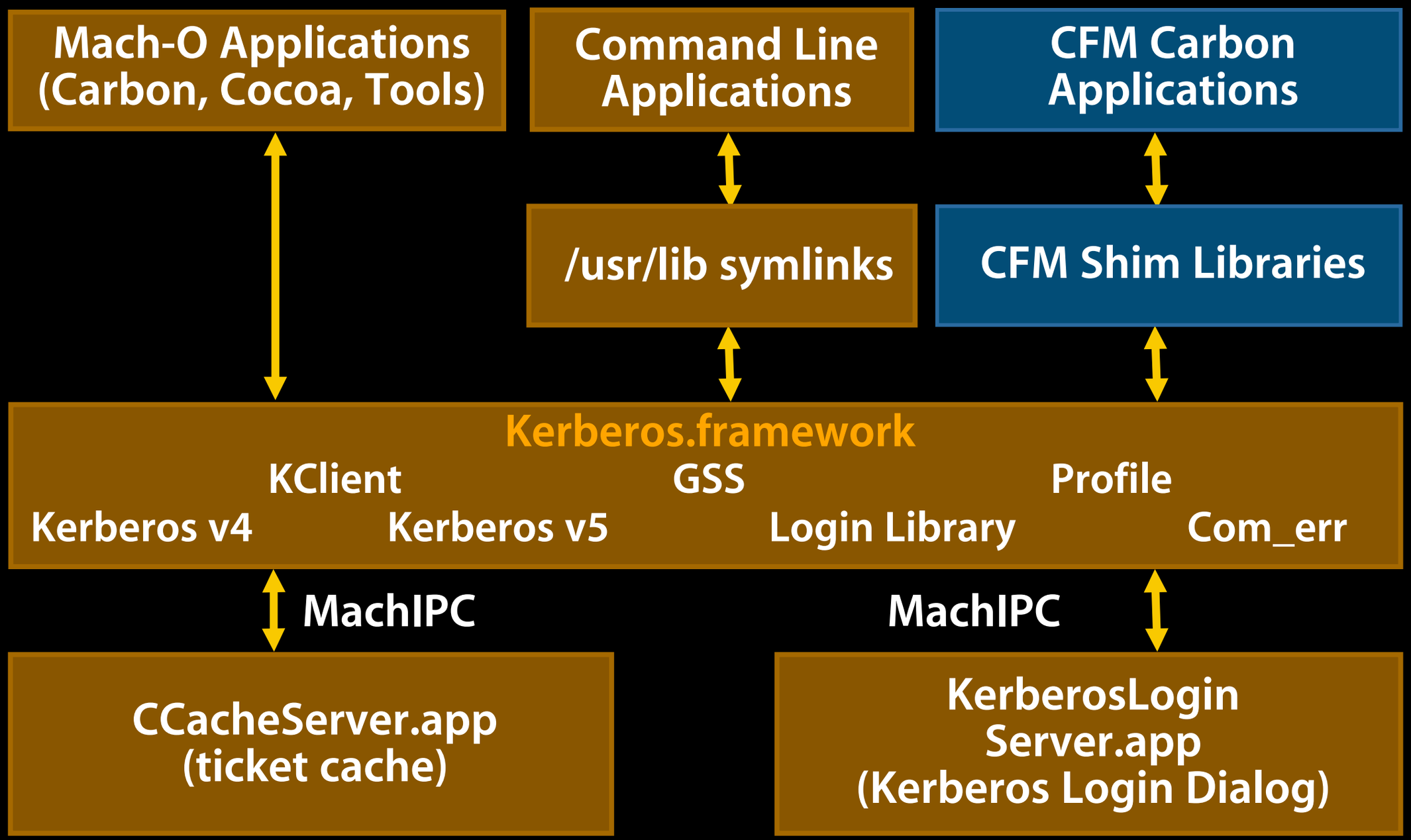
- MIT Kerberos v5 1.3
- Improved Windows interoperability
- New Kerberos Login Dialog
- UI for Addressless tickets (NAT mode)
- DNS support for KDC lookups
- Renewable Ticket Granting Tickets

Panther Kerberos

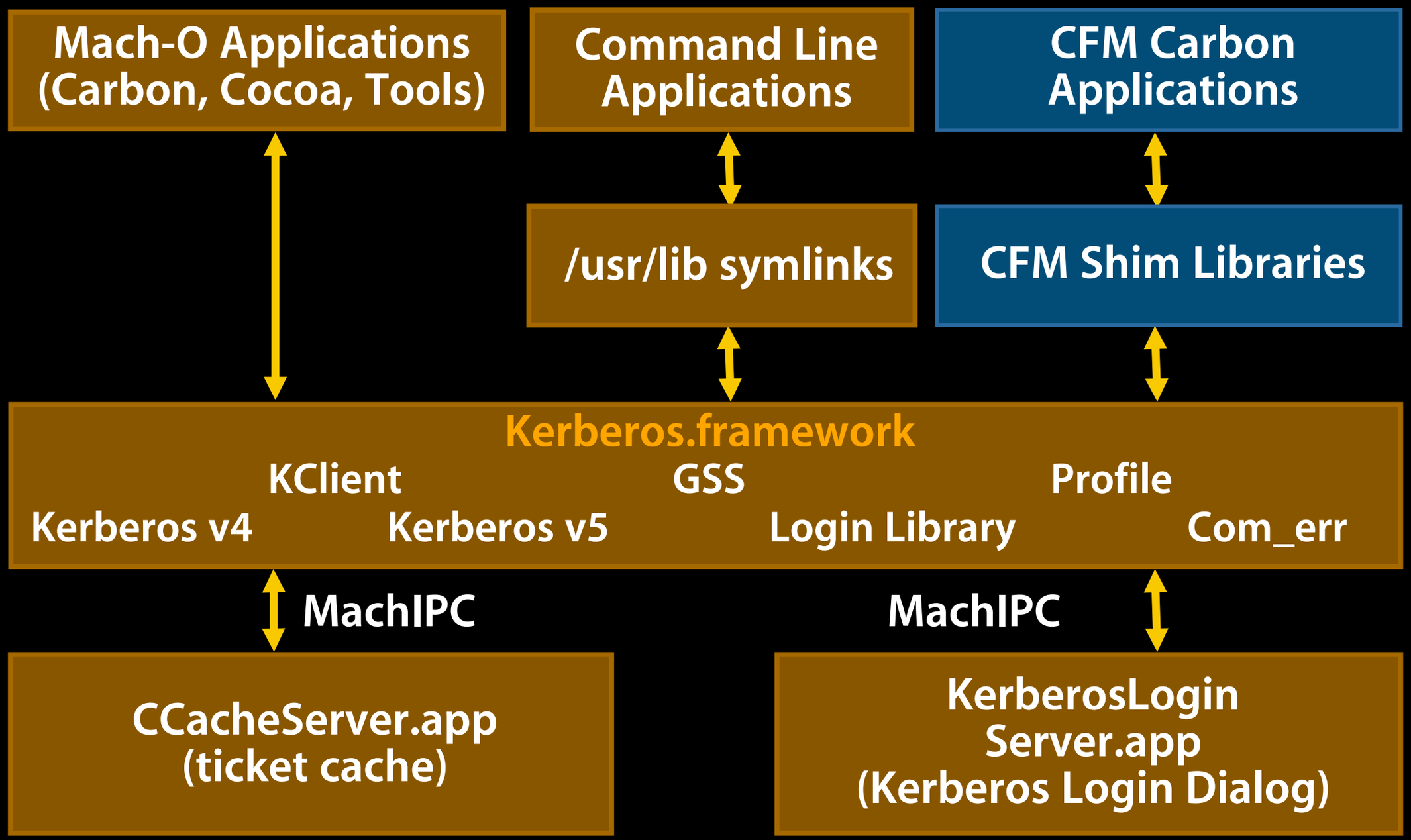
Kerberos for Macintosh 5.0

- Server components now included
  - Standard MIT KDC and related daemons
  - Common tools to operate a MIT KDC
    - kadmin
    - ktutil
- Apple includes a more integrated KDC as part of Panther Server

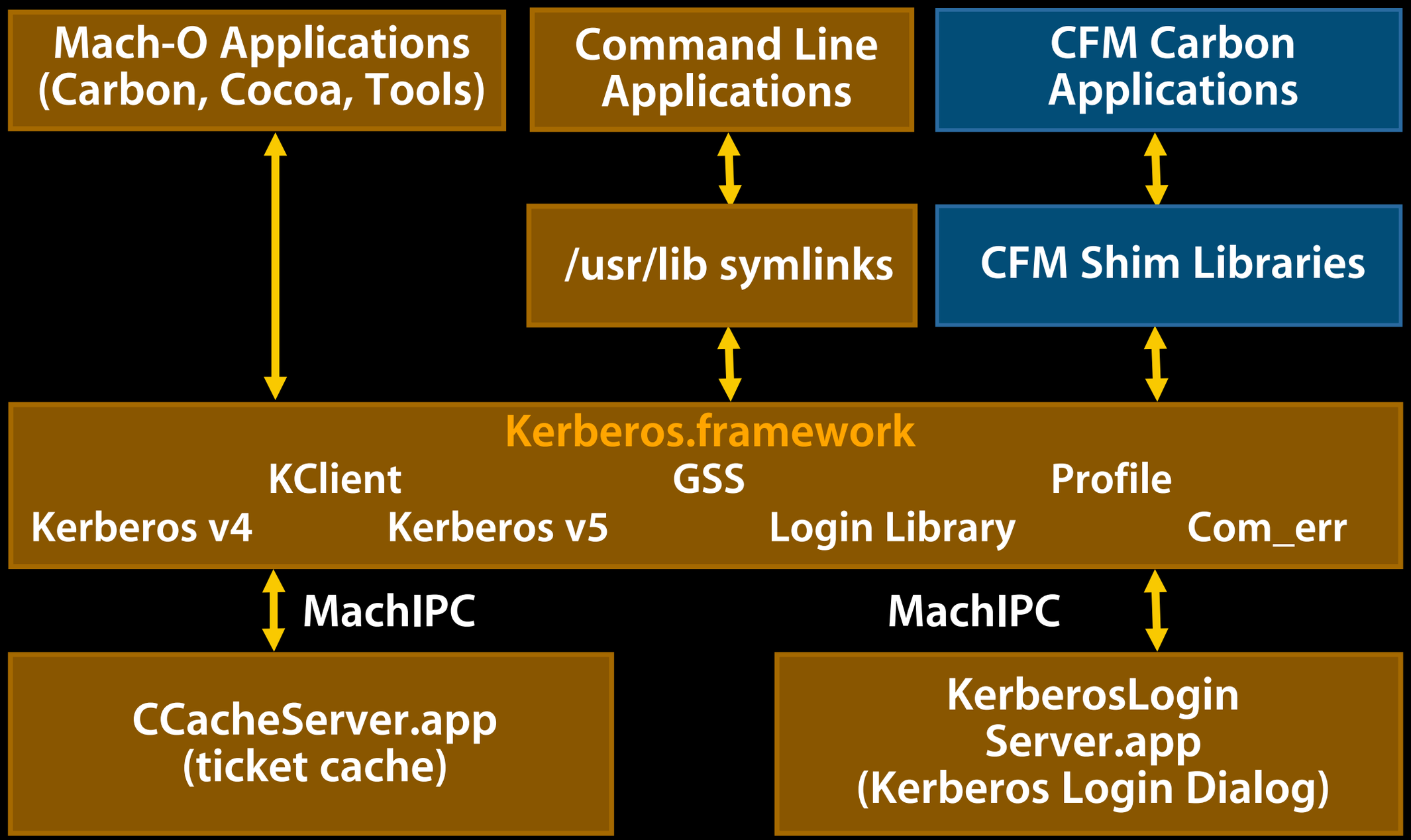
KfM 5.0 Architecture



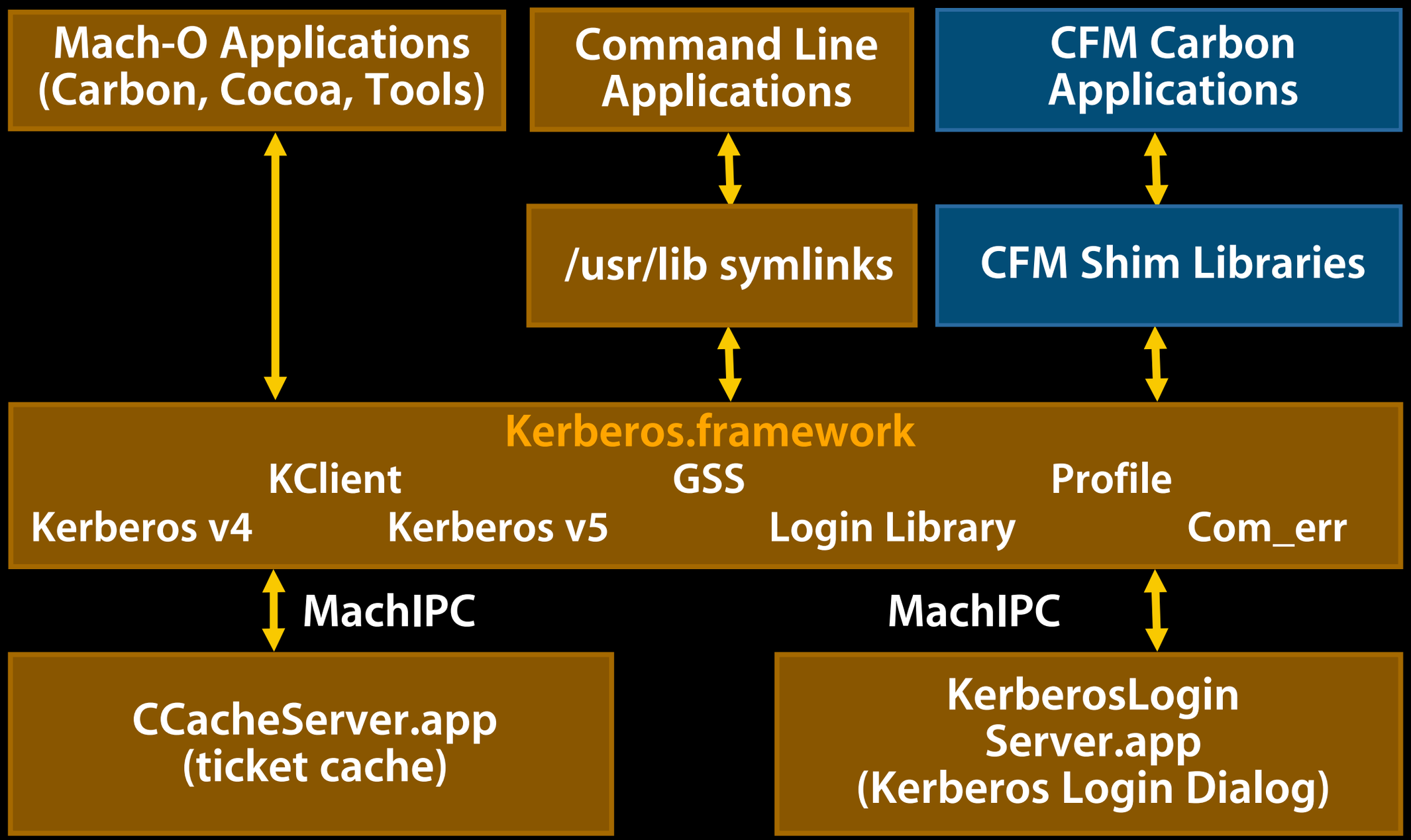
KfM 5.0 Architecture



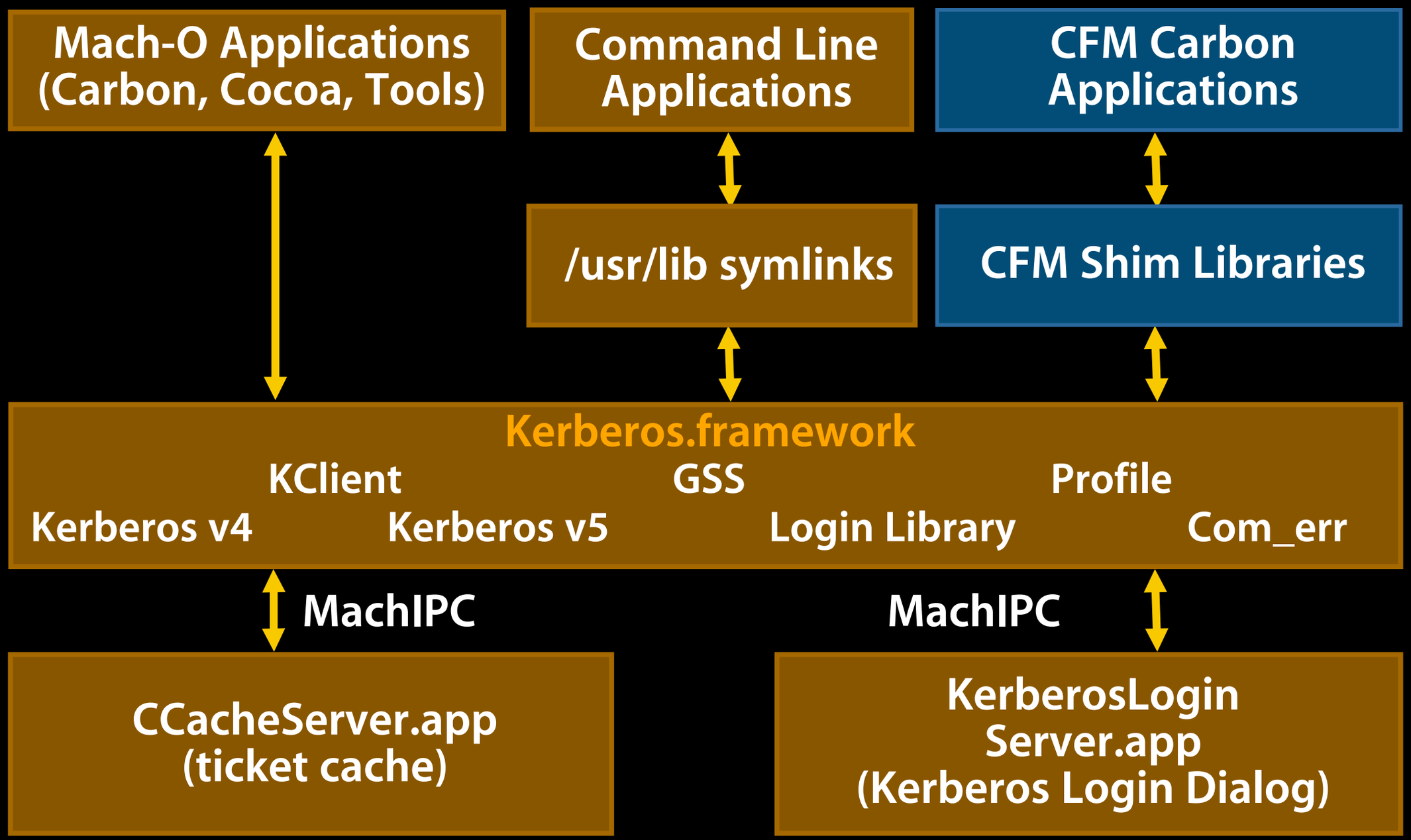
KfM 5.0 Architecture



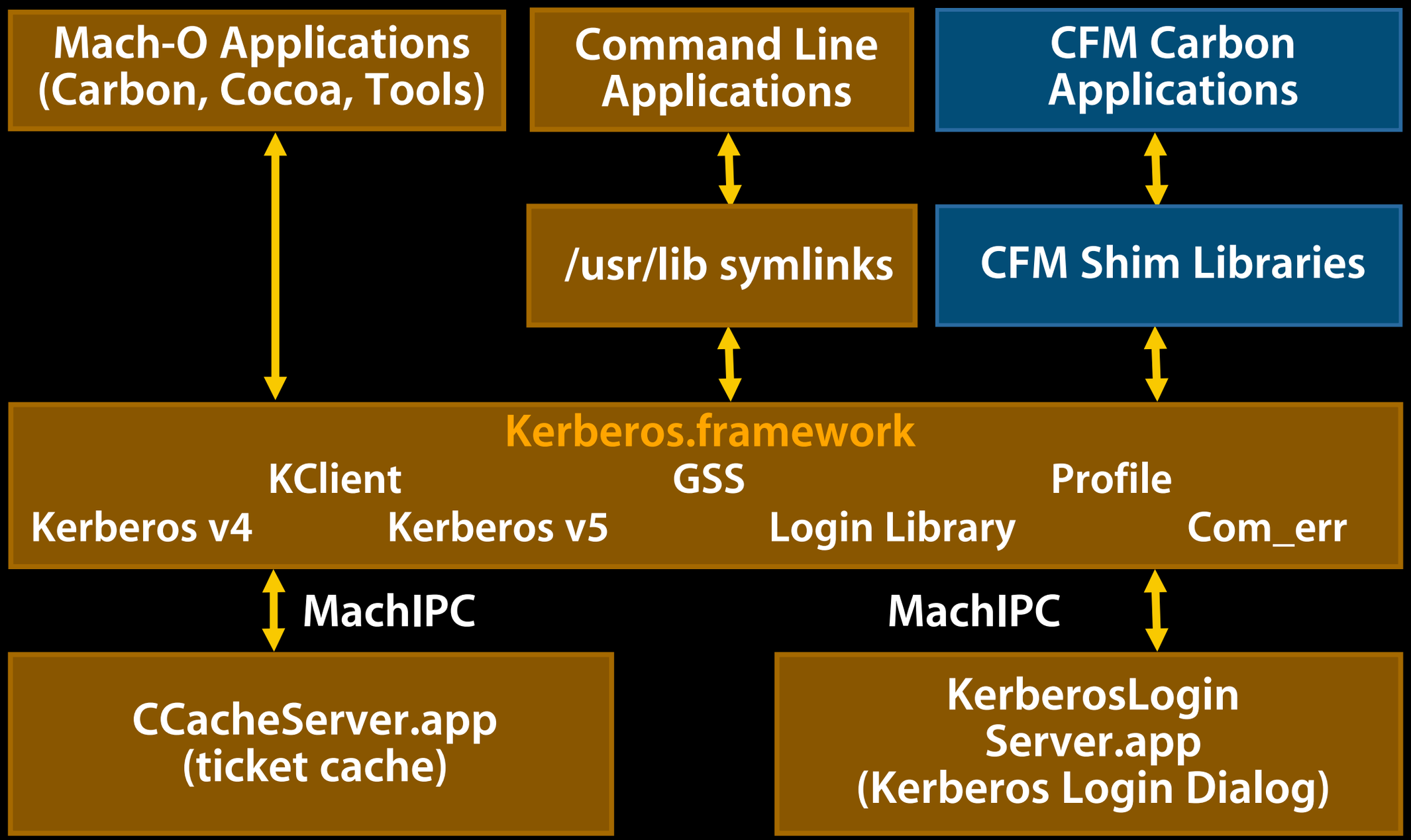
KfM 5.0 Architecture



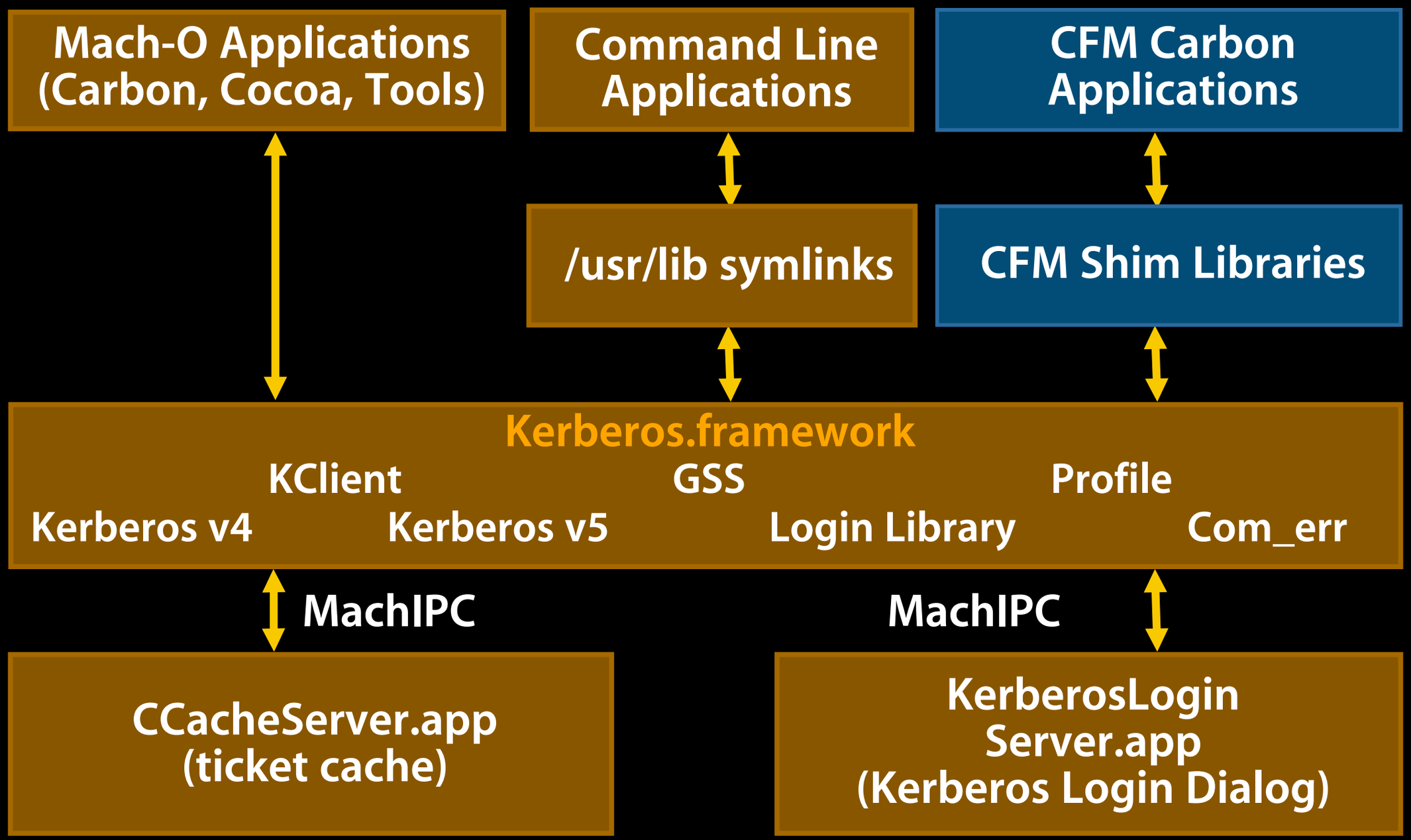
KfM 5.0 Architecture



KfM 5.0 Architecture



KfM 5.0 Architecture



Kerberos v4

- KClient 3.1
  - API compatibility libraries
  - Modern version 3 APIs
- Kerberos v4 implementation
  - Now based on MIT's core v4 compatibility library
  - Modifications to support Kerberos Login Library
- Applications can use either API for v4 calls
  - Should not develop new code for v4

Kerberos v4

Why you should no longer use it

- Only supports DES
- Cannot get addressless tickets
- Tickets do not support various feature flags
- Microsoft offers no native support for v4
- Serious security vulnerabilities
  - CERT VU #623217, 442569
- There is a solution…

Kerberos v5

- MIT Kerberos v5 1.3
  - RFC #1510: “The Kerberos Network Authentication Service”
  - Enhanced support for current IETF draft of “Kerberos Clarficiations”
- GSSAPI
  - Supports:

    RFC #2744: “Generic Security Service API: C-bindings” and
    RFC #1964: “The KerberosVersion 5 GSS-API Mechanism”

MIT Kerberos v5 1.3

- Improved Microsoft Windows interoperability
  - TCP support added to client libraries and KDC
  - RC4 crypto system
  - Microsoft set-password protocol
- Improved support for IPv6
- Initial AES support
- Builds on Darwin systems
- Many bug fixes and feature improvements

Microsoft Interoperability

- Active Directory can be used as a Kerberos v5 KDC to Mac OS X Kerberos clients
- Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) currently has no public counterpart on Mac OS X
- Security Support Provider Interface (SSPI) is very similar to GSSAPI
- TCP communication now supported
- RC4 crypto system now supported

Credentials Cache

- Bottom layer—where the tickets/credentials are stored
- Stored in memory as opposed to on disk
  - More efficient flexibility in handling multiple credentials
- Provides a cross-platform API for accessing the credentials used by v4 and v5 libraries

Kerberos Login Library

- Simplifies the process of getting Kerberos TGTs
  - Allows you to get or destroy tickets with a single call
  - Transparently handles v4 and v5 initial tickets
- Also provides API for adjusting Kerberos Login Dialog options
- Used by programs that need to perform actions on TGTs, such as administrative applications

Other Libraries

- Com_err
  - Access error codes returned by Kerberos
- Profile
  - Read information stored in the Kerberos configuration file (for example, library defaults, realms, etc.)

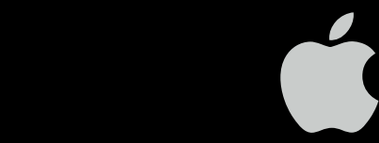
Kerberos Application

- Provides a UI for obtaining, destroying, listing, and displaying information about tickets and currently authenticated users
- Also provides a UI for setting
  - Kerberos Login dialog options
  - Favorite realm
- Dock icon presents ticket status and menu

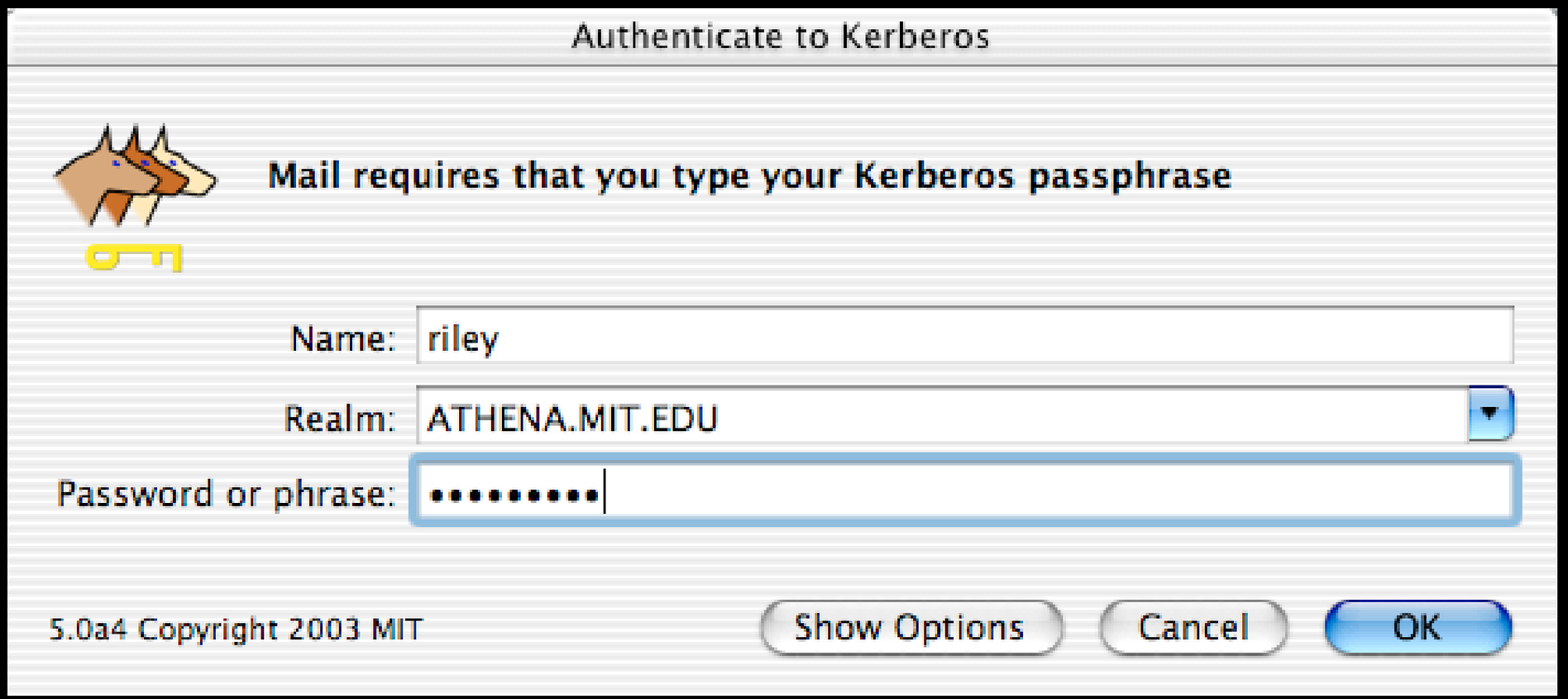
Kerberos Application

New features

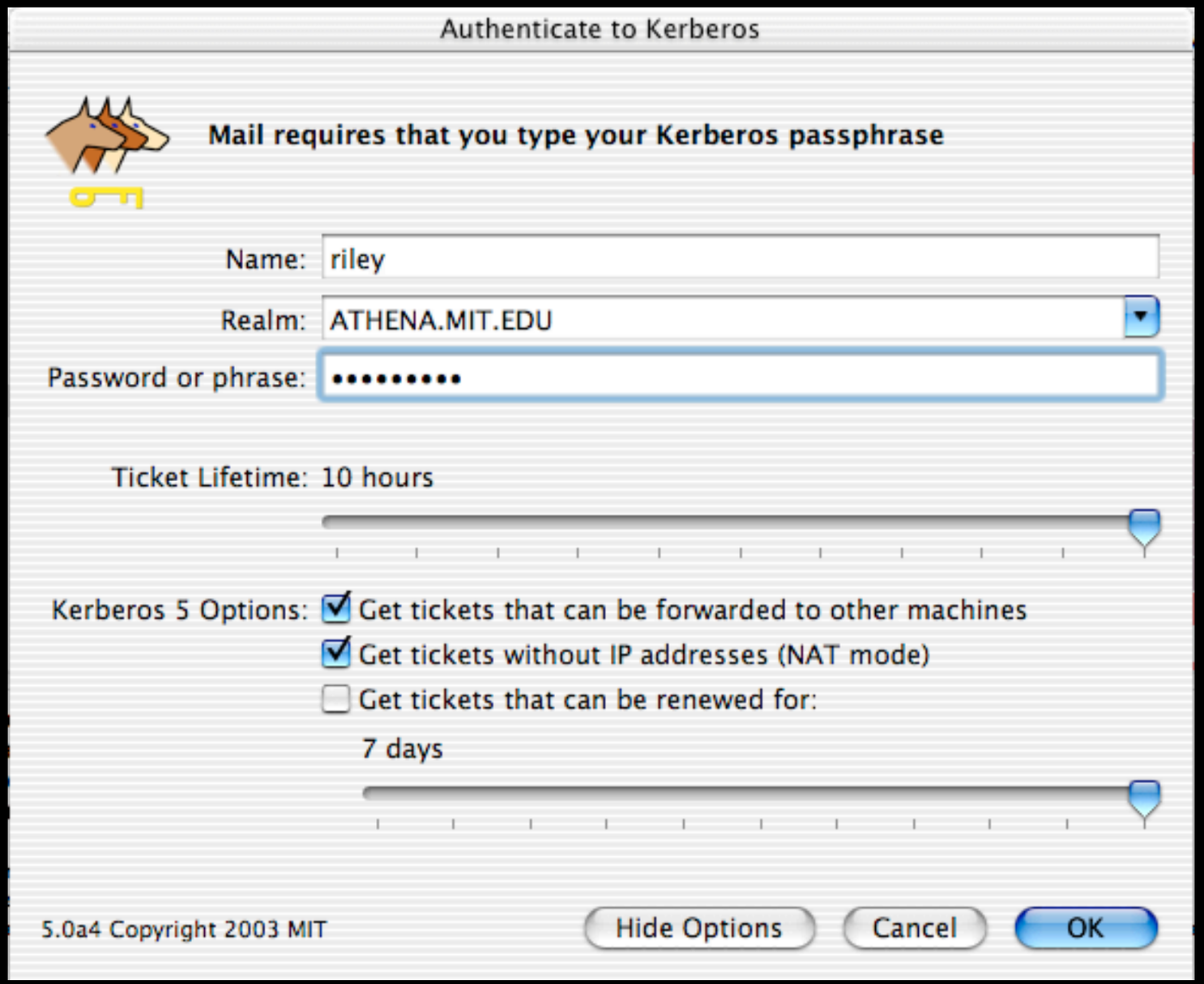
- Additions for editing new Kerberos Login Dialog defaults
- Improved performance and appearance
- Enter realms for DNS in Favorite Realms
- New time ranges for setting renewable ticket options
- Auto renews TGTs when applicable



Kerberos Login Dialog



Kerberos Login Dialog



Panther Kerberos Considerations

- Kerberos Login Dialog password field changed from MacRoman to UTF-8
- New CFM bridge libraries required
- Command line clients no longer automatically prompt when no tickets are available
- krb524 authentication requires a v4 configuration
- sudo and su now get a new empty ticket cache

Developing for Kerberos

How to Kerberize an Application

- Find the protocol specification (RFCs, etc.)
- Determine which API to use
  (KClient, Kerberos v5, GSSAPI, etc.)
- Figure out how to connect the API to the protocol
  - The data returned by the API usually needs to be massaged into a form required by the protocol (for example, Base 64)

Kerberizing FTP

- Protocol: RFC 2228, RFC 959
- Supports Kerberos v4 and GSS
  - Use KClient API for Kerberos v4
    (better support for multiple sessions)
  - Use GSSAPI for Kerberos v5

Kerberizing FTP: GSSAPI

- Session management and authentication
  - `gss_init_sec_context`
- Encryption and decryption
  - `gss_wrap`
  - `gss_unwrap`
- Connecting the API to the protocol
  - Base 64 encoding

GSSAPI Programming

Basics

- GSSAPI exchanges messages called tokens
- At the beginning of a program, GSSAPI exchanges context tokens to perform authentication and to set up encryption
- Later the program calls `gss_wrap` and `gss_unwrap` to encrypt and decrypt application data
- In general, all application data should be encrypted

GSSAPI Programming

Application choices

- Applications must select what GSSAPI security services they want
- Most applications want to explicitly turn on integrity protection, confidentiality, replay detection, and sequencing
- GSSAPI also provides a facility to forward credentials from one system to another

GSSAPI Programming

Program flow

- GSSAPI potentially requires multiple round trips for authentication
- The application does not know how many round trips are needed ahead of time
- So the application loops waiting for the authentication to finish

GSSAPI Programming

Client loop

```
input_token = NULL;
do {
  major_status = gss_init_sec_context(input_token,
output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Client loop

```
input_token = NULL;
do {
  major_status = gss_init_sec_context(input_token,
output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Client loop

```
input_token = NULL;
do {
  major_status = gss_init_sec_context(input_token,
output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Client loop

```
input_token = NULL;
do {
  major_status = gss_init_sec_context(input_token,
output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Client loop

```
input_token = NULL;
do {
  major_status = gss_init_sec_context(input_token,
output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Server loop

```
do {
  recv_token( input_token);
  major_status = gss_accept_sec_context( input_token,
      output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Server loop

```
do {
  recv_token( input_token);
  major_status = gss_accept_sec_context( input_token,
      output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Server loop

```
do {
  recv_token( input_token);
  major_status = gss_accept_sec_context( input_token,
      output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Server loop

```
do {
  recv_token( input_token);
  major_status = gss_accept_sec_context( input_token,
      output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

GSSAPI Programming

Server loop

```
do {
  recv_token( input_token);
  major_status = gss_accept_sec_context( input_token,
      output_token);
  if (output_token)
    send_token( output_token);
  if (major_status == GSS_S_CONTINUE_NEEDED)
    recv_token( input_token);
} while (major_status == GSS_S_CONTINUE_NEEDED);
```

Kerberizing Suggestions

- Use SASL if adding Kerberos to your own protocol
- Using SASL includes using GSSAPI for Kerberos v5 functionality
- GSSAPI helps isolate applications from differences in Kerberos v5 APIs
- Do not add additional Kerberos v4 support

Summary

Points to Remember

- Kerberos is a simple yet power solution to authentication and single sign on
- Provides an unobtrusive authentication experience to the user
- Available on a wide variety of platforms
- Sites should be migrating from Kerberos v4
- Panther and Panther Server will provide a wealth of Kerberized services

Reporting Bugs

- File bugs about Kerberos in Mac OS X with Apple

Wrap Up

Roadmap

101 Security: Architectural Overview The Security Infrastructure in Mac OS X	**The Marina** **Mon., 5:00pm**
607 Authentication in Mac OS X Server Authentication in Mac OS X and Mac OS X Server	**Nob Hill** **Tue., 5:00pm**
610 Directory Services Open Directory for Mac OS X and Mac OS X Server	**The Mission** **Wed., 10:30am**
FF016 Security Feedback Forum Tell Us What You Think	**North Beach** **Fri., 2:00pm**

Who to Contact

Craig Keithley
Security Technology Evangelist
keithley@apple.com

Marshall Vale
Project Manager Kerberos Development
Massachusetts Institute of Technology
mjv@mit.edu

Skip Levens
Server Technology Evangelist
slevens@apple.com

http://developer.apple.com/wwdc2003/urls.html

For More Information

Reference Library

Documentation

MIT Kerberos

http://mit.edu/kerberos/www/papers.html

MIT Kerberos for Macintosh

http://mit.edu/macdev/www/kerberos.html

Reference Library

Online Resources

KB107153: Mac OS X 10.2 About Using Kerberos

http://docs.info.apple.com/article.html?artnum=107153

Windows 2000 Kerberos Interoperability and Authentication

http://www.microsoft.com/windows2000/techinfo/howitworks/security/kerbint.asp

Answers to Frequently Asked Kerberos Questions

http://support.microsoft.com/default.aspx?sd=msdn&scid=kb;en-us;266080

Reference Library

Online Resources (continued)

Describing GSSAPI in terms of SSPI

http://msdn.microsoft.com/library/en-us/dnsecure/html/sspikerberos.asp

IMAP4 Authentication Mechanisms

http://www.faqs.org/rfcs/rfc1731.html

Internet RFCs

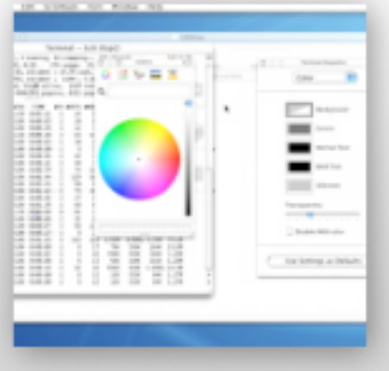
http://www.ietf.org/rfc.html

Source Code

Darwin Kerberos Page

http://developer.apple.com/darwin/projects/kerberos/

Q&A



Craig Keithley
Security Technology Evangelist
keithley@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference